Two hours

SECTION A AND SECTION B ARE COMPULSORY

UNIVERSITY OF MANCHESTER
SCHOOL OF COMPUTER SCIENCE

Fundamentals of Databases

Date: Wednesday 27th January 2016

Time: 09:45 - 11:45

The Paper is in THREE Sections

You must answer Section A, worth 10 marks
You must answer Section B, worth 10 marks
You must answer ONE of the two questions in Section C, worth 20 marks.

Use a SEPARATE answerbook for each QUESTION.

This is a CLOSED book examination

The use of electronic calculators is NOT permitted

Section A

This section contains one question.
You MUST answer it using a SEPARATE answerbook.

1. This question has three parts, from *1.a* to *1.c*, and they may have subparts in turn.

   a) Briefly explain in what sense the data definition language (DDL) and the data manipulation language (DML) in a DBMS have analogous roles by stating what is common and what is different between them. (2 marks)

   b) Assume the following (partially-specified) relational schema with three relations *R*, *S*, and *T*:

```
R(a:int, b: int, c: str)
S(d:int, e: int, a: int)
T(d:int, g:str, h:int)
```

   and the following database state:

```
R = {(5,0,'x'), (2,1,'y'), (12,1,'x')}
S = {(101,43,2), (87,34,2), (98,24,5)}
T = {(101,'Tim',4),(101,'Jim',3),(87,'Jan',18)}
```

   Assume that the primary keys of *R*, *S*, and *T* are, respectively, $\{a\}$, $\{d\}$ and $\{d,g\}$. Further assume that $T.d$ references $S.d$ and that $S.a$ references $R.a$.

   i) Explain what constraint would be violated by executing the following SQL statement: `UPDATE R SET b = 'No';` on the above database state. (1 mark)

   ii) Explain what constraint would be violated by executing the following SQL statement: `DELETE FROM R WHERE a = 5;` on the above database state. (2 marks)

   iii) The fact that the primary key of *T* includes a foreign key (viz., the primary key of *S*) suggests that, in terms of entity-relationship modelling notions, $T.g$ is what kind of attribute? (2 marks)

   c) Assume the relational schema and database state given earlier in this question, then answer the following:

   i) State in English what the following SQL statement retrieves:

```
SELECT   s.d AS s_key, count(t.g) AS count
FROM     S AS s, T AS t
WHERE    s.d = t.d
GROUP BY s.d
```

   (1 mark)

   ii) Write out the result set (i.e., the set of tuples) returned by evaluating the query just given on the database state given earlier in this question. (2 marks)

Section B

This section contains one question.
You MUST answer it using a SEPARATE answerbook.

2. This question has four parts, from *2.a* to *2.d.*

   a) Enumerate and explain the three different approaches for making SQL computationally complete discussed in the course..

   (3 marks)

   b) Enumerate the three most common problems caused by concurrency in database management systems (as studied in this course unit) and, for each problem, state what happens when the problem manifests itself and no concurrency control protocol is in use.

   (3 marks)

   c) Briefly describe the role that functional dependencies play in the process of schema normalization and illustrate your answer by showing (informally) how the functional dependencies F (below) would be used to normalize the SC schema (below):

   $SC(studentNumber, studentName, courseNumber, courseName, mark)$

$$
\begin{array}{rcl}
F = \{ & & \\
studentNumber & \rightarrow & studentName, \\
courseNumber & \rightarrow & courseName, \\
studentNumber, courseNumber & \rightarrow & mark \\
\} & &
\end{array}
$$

   (2 marks)

   d) The physical organization of data in a file into records and pages in secondary storage affects query execution performance. Briefly explain why and illustrate your answer with an example. (2 marks)

Section C

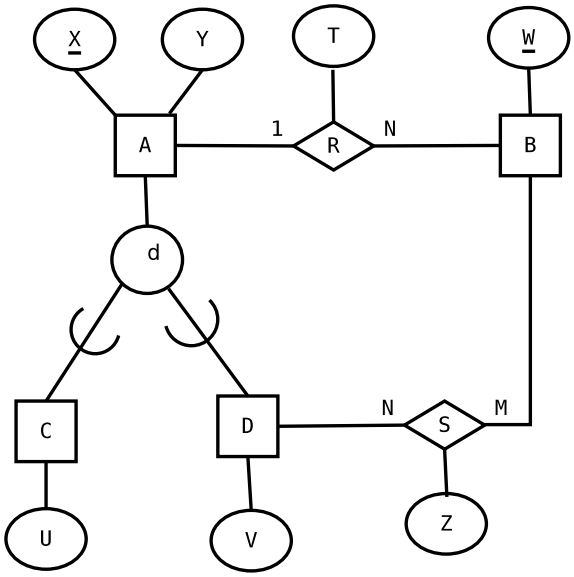
Answer one of the two questions in this section.
Use a SEPARATE answerbook for the question you choose.

3. This question has three parts, from *3.a* to *3.c*.

   a) Apply the ER-to-relational mapping procedure that you were taught in the course and show the stepwise derivation of the relational schema corresponding to the following ER model, step by step.

   Note that some steps may not be applicable, in which case you should note that fact. Recall that a later step sometimes modifies the outcome of a previous step, in which case you should show the modifications to the previous outcome.

   (10 marks)



b) Assume that you are a member of a team of database designers. A colleague of yours had a set of interviews for the purpose of data requirements elicitation with a customer representative. The *text in italics below*, represents the result of her data gathering exercise. It has been noted, however, that another visit to the customer representative is needed to gather more information before one can draw an ER diagram that captures the requirements. Write up to five questions (as specific and focussed as possible) the answers to which would allow you to draw the ER diagram.

Remember not to simply make an assumption (based on common sense) that you know in advance what is the answer that the customer would give. Note that your questions must have the specific purpose of eliciting the missing information to draw an ER diagram.

PVPP is a specialist delivery business using vans to deliver packets to specific pick-up points.

- *They store data about packets, namely, for each packet, its unique ID, its volume (say, in cm^3), its weight, and its value.*
- *They store data about vans, namely, for each van, its unique license number, its volume capacity (say, in cm^3), and its weight capacity.*
- *They store data about pick-up points, namely, for each pick-up point, its postcode, its opening time and its closing time.*
- *Each packet is assigned to one and only one van, whereas a van can be assigned many packets.*
- *Every van must deliver to a pick-point, but a pick up point may not have vans that deliver to it.*

(5 marks)

c) Assume that your questions in the previous part have been answered and that the answers have allowed the ER diagram to be drawn and mapped onto a logical model.

Further assume that part of the SQL script that was used to create the corresponding database contains the following lines:

```
CREATE TABLE van (
       LicenceNumber int,
       volume_capacity int,
       weight_capacity int,
       primary key (LN));

CREATE TABLE packet (
       ID int,
       volume int,
       weight int,
       value int,
       LicenceNumber int,
       primary key (ID),
       foreign key (LicenceNumber) references van(LicenceNumber));
```

Write a SQL query against the two tables above that lists the licence number of any van for which the weight of all the packets assigned to it exceeds that van's weight capacity. (Do not worry about the volume capacity of the van here.) (5 marks)

4. This question has five parts, from *4.a* to *4.e*.

   a) Consider a database that stores sensor-collected data. The data describes sites in the city (e.g., roads) where traffic data was collected, the specific lane of the road where vehicle counting is performed (i.e., `lane_id`), the direction associated with the lane (e.g., north, east, etc.), each detected vehicle (including its class information, such as car, motorcycle, bus, etc.), specific dates and time periods in which each vehicle was detected (note that `date_flag` distinguishes holidays and weekends from working days) as well as the vehicle's speed, as described below.

   Does the table below represent a good design? Illustrate your answer with an example of a table update problem that might result from the current design of the table.

```
CREATE TABLE sensor_recording (
             site_id SMALLINT NOT NULL,
             site_name VARCHAR(15),
             lane_id SMALLINT NOT NULL,
             lane_name VARCHAR(15),
             direction_id SMALLINT,
             direction_name VARCHAR(15),
             vehicle_id SMALLINT NOT NULL,
             vehicle_class_id SMALLINT,
             vehicle_length DECIMAL(2,1),
             vehicle_class_name VARCHAR(15),
             year_month_day DATE NOT NULL,
             time_day TIME NOT NULL,
             vehicle_speed DECIMAL(2,1),
             date_flag SMALLINT,
             date_flag_text VARCHAR(15),
             traffic_volume INT,
PRIMARY KEY (site_id, lane_id, vehicle_id,
             year_month_day, time_day));
```

(2 marks)

   b) If the above schema is in 3NF or BCNF, then argue why this is so from the respective definitions. If it is not in 3NF or BCNF, then normalize it step-by-step to 3NF or BCNF.

(5 marks)

c) Consider the following tables. Note the table named `number_of_bookings`, which contains aggregates (i.e., the number of bookings for a particular flight) that are derived from the `booking` table. To keep the `number_of_bookings` table up to date, the `booking` table must be monitored, so that these changes can be reflected on the `number_of_bookings` table. Write three triggers that respond to insert, delete and update events on the `booking` table and propagate changes to `number_of_bookings` accordingly.

```
CREATE TABLE booking (
        passenger_id VARCHAR(10)
          REFERENCES passenger(passenger_id) ON DELETE CASCADE,
            flight_number VARCHAR(10)
          REFERENCES flight(flight_number),
        year_month_day_time  DATE,
PRIMARY KEY (passenger_id, flight_number, year_month_day_time));

CREATE TABLE number_of_bookings (
            flight_number VARCHAR(10)
          REFERENCES flight(flight_number),
        year_month_day_time  DATE,
        num NUMBER,
PRIMARY KEY (flight_number, year_month_day_time));
```

(6 marks)

d) Consider the schedule below. Note that *balance_acc_x* and *balance_acc_y* represent the balance of bank accounts *acc_x* and *acc_y*, respectively. State whether the use of locks ensures serializability. Explain your conclusion by assuming that, when execution starts, *balance_acc_x = 100* and *balance_acc_y = 400* and showing that the schedule is (or is not) serializable.

Time	Transaction_1	Transaction_2
t1	begin_transaction	
t2	write_lock(balance_acc_x)	
t3	read(balance_acc_x)	
t4	write(balance_acc_x + 100)	
t5	unlock(balance_acc_x)	
t6		begin_transaction
t7		write_lock(balance_acc_x)
t8		read(balance_acc_x)
t9		write(balance_acc_x * 1.1)
t10		unlock(balance_acc_x)
t11		write_lock(balance_acc_y)
t12		read(balance_acc_y)
t13		write(balance_acc_y * 1.1)
t14		unlock(balance_acc_y)
t15		commit
t16	write_lock(balance_acc_y)	
t17	read(balance_acc_y)	
t18	write(balance_acc_y - 100)	
t19	unlock(balance_acc_y)	
t20	commit	

(4 marks)

e) Consider databases of large sizes, containing, for example, hundreds of relations, each with millions of tuples and hundreds of attributes. Briefly explain what help indices provide and suggest the type of index that would be particularly useful in the above cases.

(3 marks)

END OF EXAMINATION